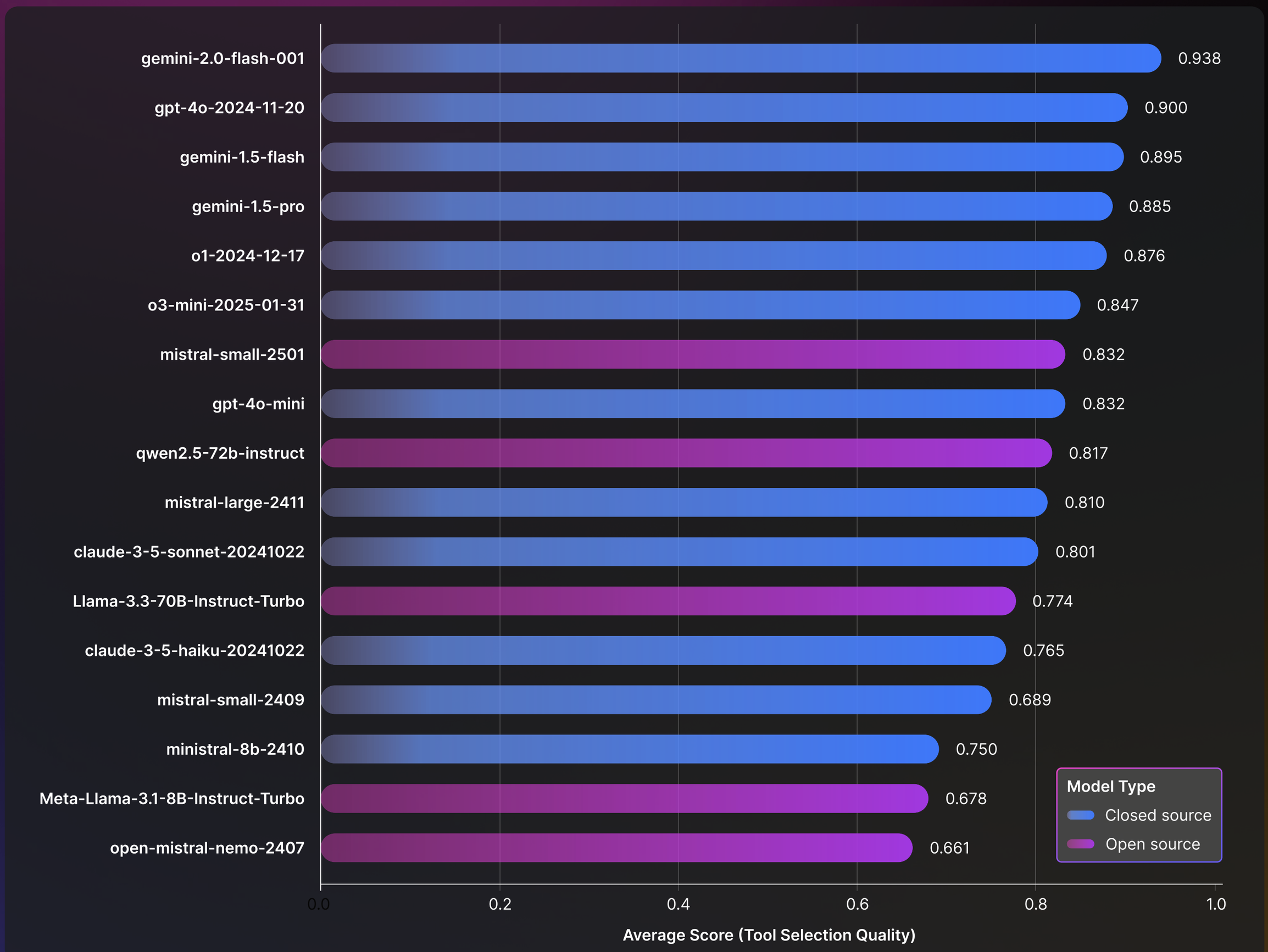
Agent Leaderboard

# Which LLMs Work Best for AI Agents?



Galileo

# Top LLMs For AI Agents




Galileo

# Agent Leaderboard on HuggingFace

## Leaderboard

https://huggingface.co/spaces/galileo-ai/agent-leaderboard

| 17 | 14 | TSQ |
|---|---|---|
| **Total Models** | **Evaluation Datasets** | **Evaluation Metric** |
| 12 Private | Multi-domain Testing | Tool Selection Quality |
| 5 Open Source | Real-world use cases | GPT-4o Based Judge |

Tool calling performance on 14 datasets

Galileo

# Performance Champion

**Gemini-2.0-flash dominates** with a 0.938 score at a **very** affordable cost, excelling in both complex tasks and safety features.

01

# Overpriced Models

02

The top 3 models span **10x price** difference with only a 4% performance gap - some models are over-priced for their value for AI agents.

# Open vs. Closed Source

03

**Mistral-small-2501 leads in open source models** and performs similar to GPT-4o-mini at 0.83, signaling OSS maturity in tool calling.

# DeepSeek Misses

DeepSeek V3 and R1 were among the models that **failed to make the rankings** due to limited function support.

04

# Dataset Overview

## Tool Selection | Parallel Calling | Edge Cases

| | Samples | Category | Dataset Name | Purpose |
|---|---|---|---|---|
| Single-Turn | 100+100 | **Single Function Call** | xlam_single_tool_single_call, xlam_multiple_tool_single_call | Evaluates basic ability to read documentation and make single function calls |
| | 200 + 50 | **Multiple Function Call** | xlam_multiple_tool_multiple_call, xlam_single_tool_multiple_call | Tests parallel execution and result aggregation |
| | 100 | **Irrelevant Query** | BFCL_v3_irrelevance | Tests ability to recognize when available tools don't match user needs |
| | 100 | **Long Context** | tau_long_context | Assesses handling of extended interactions and complex instructions |
| | 100 | **Missing Function** | xlam_tool_miss | Tests graceful handling of unavailable tools |
| Multi-Turn | 50 + 30 | **Single Function Call** | BFCL_V3_multi_turn_base_single_func_call, toolace_single_func_call | Tests basic conversational function calling abilities |
| | 50 | **Multiple Function Call** | BFCL_v3_multi_turn_base_multi_func_call | Evaluates handling of multiple function calls in conversation |
| | 100 | **Missing Function** | BFCL_v3_multi_turn_miss_func | Tests graceful handling of unavailable tools |
| | 100 | **Missing Parameters** | BFCL_v3_multi_turn_miss_param | Assesses parameter collection and handling incomplete information |
| | 100 | **Composite** | BFCL_V3_multi_turn_composite | Tests overall robustness in complex scenarios |

# Implications

Galileo

## Implications for Agent Development

### Choose Models Based on Task Complexity

01 Simple tasks work with most models. Complex workflows requiring multiple tools need models with 0.85+ scores in composite tests.

### Plan for Error Handling

02 Models with low tool selection scores need guardrails. Add validation layers and structured error recovery, especially for parameter collection.

### Consider Context Management

03 Long conversations require either models strong in context retention or external context storage systems.

### Selecting Reasoning Models

04 While o1 and o3-mini excelled in function calling, DeepSeek V3 and R1 were excluded from rankings due to limited function support.

### Implement Safety Controls

05 Add strict tool access controls for models weak in irrelevance detection. Include validation layers for inconsistent Performers.

### Open vs Closed Source

06 Private models lead in complex tasks, but open-source options work well for basic operations. Choose your model based on your scaling needs.

# What Makes Tool Calling Difficult?

Galileo

## Tool Selection Dimensions

### Precision

Accuracy in selecting appropriate tools and avoiding incorrect or redundant tool calls

### Recall

Identifying and using all necessary tools for completing the task

### Redundancy

Avoiding unnecessary tool calls when information is already available

## Argument Complexity

### Parameter Precision

Correct naming and value assignment for tool arguments

### Completeness

Including all required arguments and proper handling of optional parameters

## Decision Scenarios

### Should Call Tools

- Calls correct tools → Success
- Doesn't call tools → Failure
- Calls partial tools → Recall issue

### Should Not Call Tools

- Doesn't call tools → Success
- Calls tools → Redundancy issue
- Information in history → No tools needed

# Get the Results and More Insights

Leaderboard

https://huggingface.co/spaces/galileo-ai/agent-leaderboard

Galileo